

PROJECT

OBJECTIVES:

The project will give you the opportunity to review and practice on the following:

- Types of Data structure: Array Based Structure, Linked List structures, Hashed Data Structure or Binary Search Tree Data structure
- Condition structure, loop structure
- read from keyboard
- working with the file: -file: open, check file existing, read file, write to file
- class, object: constructor, mutator methods, accessor methods, toString and others, and how to access the class members from client site
- display information on screen with requested format on numbers
- Handling input errors
- Write comments

REQUIREMENT:

At the school ABC, the information of students are stored in two files:

- the file named **studentInformaiton.xlsx** that includes: student id, last name, first name, Security Number, birth day, phone number
- the file named **studentGrade.xlsx** that include student id, and all the classes that students completed with the letter grade and the classes that students are currently taking

Create an application for this School System that allows the Administration Office to work with students information.

First, define an object of data structure type then read file studentInformation.csv and studentGrade.csv; for each line of them, create a node of Student object, then insert the node to the above data structure

You can select any kind of data structures to store all nodes and to work on this project but you only choose one of data structures we have learned during semester (DO NOT SELECT THE ONE FROM JAVA LIBRARY).

HOW TO GET File studentInformation.csv and file studentGrade.csv

-download from ecampus file studentInformation.xlsx and file studentGrade.xlsx

-store them to your folder with the same extension

-Open file .xlsx then SAVE AS with the same name but different extension .csv (DO NOT JUST CHANGE EXTENSION, YOU HAVE TO SAVE AS WITH NEW EXTENSION)

Then, display the following tasks to allow users to select

SCHOOL ABC

1. Add One New Student from the keyboard (
2. Remove One Student
3. Find one Student by ID
4. Add a class for one student
5. Drop a class for one student
6. Print the list of student names in one class
7. Print out the transcript of one student
8. Show all studdents in the data structure

After finishing one task, the program will loop back to allow users do other tasks until users want to exit.

TASK 1: Add One New Student from the keyboard (INSERT) This task helps to insert one new student to the data structure. The information of one student are read from the keyboard.

TASK 2: Remove One Student (DELETE): This task helps to remove one student from data structure. Ask for an ID, for example, 12345, remove student and print out the message: "Student with Id 12345 is removed" OR "Student with id 12345 is not in the system"

TASK 3: Find a student by ID (FETCH) (for example id = 12345); the program will ask for an id that is entered from the keyboard, then display the information of student found as below where if the class with grade 'X' print "Not Complete" instead of the letter grade;
If cannot find the id then display the message "ID cannot be found"

```
Student:      James Smith
Student ID:   12345
SS Number:   123456789
Birthday:    03/21/1996
Phone:       2147256767
Address:     123 Walnut rd Dallas TX 74243
Classes:
FA2016-MATH1325 - A
FA2016-PHYS1401 - B
SP2017-COSC1301 - A
SP2017-MATH1414 - B
SP2018-HUMA1302 - A
SP2018-GOVT2305 - A
SP2017-COSC1301 - A
SP2017-ENGL1301 - B
SP2018-MUSI1181 - Not Complete
SP2018-MATH1414 - Not Complete
```

TASK 4: Add a class for one student

The program will ask for the id and the class name, then the program will fetch the student with the id, then add the new class to the class list where the grade of the new class is 'X'

TASK 5: Drop a class for one student

The program will ask for the id and the class name to drop.

The program will fetch this student with id then remove the class from the list class. If the class has the grade that is not X, then display the message "The class is completed, cannot be dropped" otherwise "Drop class successfully"

TASK 6: Print the list of students in one class (Search all student in one class): the program will ask for the class name. It should be in the format "semester (2characters) year (4digits) – 4 letter of class 4 digits of class". For example: "FA2016-COSC2436", then use the entered class name as a key to search in data structure.
If it is found, print the id and first name and last name of the student in the following format. Continue to search until reaching the last node on the data structure. The final output of this task:

```
LIST OF STUDENT IN CLASS SP2018-MATH1414
12345      James Smith
54321      Johnson Kennedy
```

TASK 7: Print out the transcript of one student. Ask for the id from the keyboard of the student that you want to print out the transcript. Search the student in the data structure by using id as a key.
If the student is found, display the transcript on the screen in the following format

```
SCHOOL ABC - TRANSCRIPT
```

COSC2436 – Liam Le

```

-----
Student:      Johnson Kennedy
Student ID: 54321
Date:        10/ 07/ 2018
-----
FA2016-MATH1325 - A
FA2016-PHYS1401 - B
SP2017-COSC1301 - A
SP2017-MATH1414 - B
SP2018-HUMA1302 - A
SP2018-GOVT2305 - A
SP2017-COSC1301 - A
SP2017-ENGL1301 - B
SP2018-MUSI1181 - Not Complete
SP2018-MATH1414 - Not Complete

```

TASK 8: Show all (SHOW ALL)

Just call show all method to display all the student in the structure

TASK 0: //exit

Before terminate the program, do the following

- Open the file "studentInformation.csv" to write
- Open the file "studentGrade.csv" to write
- Use the logic show all of the structure to read each not information then write to file in the following format:

WRITE TO FILE studentInformation.csv:

"Id, last name, first name, ss number, phone, address"

For example:

12345,Smith,James,123456789,3/21/1996,2147256767,123 Walnut rd Dallas TX 74243

WRITE TO FILE studentGrade.csv:

"id, className1, grade1, className2, grade2, etc.."

For example:

54321,SP2017-COSC1301, A,SP2017-ENGL1301,B,SP2018-MUSI1181,X,SP2018-MATH1414,X

Remember to close the file before terminating the program

HOW TO DO THE PROJECT

- Read the requirement to understand what the requirement asks for then decide how many data type classes you need for the program and draw UML for each one
- Read the requirement to understand the logic then write the pseudo-code that lists what you suppose to do in main()
- Input file studentInformation.csv and file studentGrade.csv: downloaded the file studentInformation.xlsx and studentGrade.xlsx from the page Project on eCampus, the save them with extension .csv

NOTES:

- You can select any data structure to handle the nodes of Students **but do not use the one from java library**
- Suggestion: it is easy if you define one class, for example, class Aclass that keeps the class name and the letter grade as data members. If the class does not have the grade yet, the grade get the value 'X'
- Suggestion: Define a Java ArrayList to hold all classes Information. Then, this list of class will be come a data member of Student
- To review: how to open file, read or write, how to format nice output, or all anything relate to the project, you can check the answer in the page "Learn From Question" of contact me via e-mail or stop by my office hours
- The project name should be: SP2018PROJECT_YourLastName
- The driver class should be: SP2018PROJECT_StudentService.java

-The data type should have the name start with SP2018PROJECT , for example, class SP2018PROJECT_AClass_yourLastName, etc.

HOW TO TURN IN

- SP2018PROJECT_Design.docx contains UML of each data type class and pseudo-code
- SP2018PROJECT_StudentService_yourLastName.java (driver class)
- SP2018PROJECT_StudentService_yourLastName.class
- All data type classes and their file .class

HOW TO GRADE THE PROJECT

Turn in the project on time	10
Submit all files that need to run your project	2
compile success with all the requirements	10
UML of data type classes and psuedo-code	5
Write comments	5
class Data structure (you can choose any data structure type we learned) that hold all the tasks we need for the project	
Data type class to hold information of Students that hols all the tasks relate to students	5
Data type class AClass to hold the class name, grade and all action one class	2
Create data structure	1
Read 2 files, create the nodes, insert to the data structure	5
Display menu and handle to loop back	2
TASK 1 – Add one new student from keyboard	2
TASK 2 - remove one student	2
TASK 3 – read information of one student	2
TASK 4 – add a class for one student	5
TASK 5 – drop a class for one student	5
TASK 6 – print list of students in one class	5
TASK 7 – print transcript of one student	5
TASK 8 - show all nodes	2
TASK 0: open 2 files to write – write success in corerct format	5
Project scores	80